Jayus Solutions
Project Report

James Mauch, Keith Lahmann, John Chamblee, Adiessa Bell

Capstone Project Spring 2022

The University of West Florida

Apr 29, 2022

CIS4595 202201 Capstone

Dr. Owsnicki-Klewe

Executive Summary

StudyIsland is a web app and social media platform designed to bring the power and efficiency of study groups to everyone in the comfort of their homes. We will provide utilities for forming study groups (Islands), sharing information, and talking about shared interests. Individual islands will be represented using a message board system that will allow users who are a part of that island to share information related to the subject that island is focused on. Within the islands, users will be able to form study groups of varying size where they can discuss information in real time.

Table of Contents

Table of Contents

List of Figures

1 Final Requirements and Initial Requirements Comparison

Project requirements have remained mostly consistent throughout the course of development. There has not been a need to change the tech stack in order to meet requirements based on our initial use cases and user stories. Testing requirements were changed as we moved away from test driven development due to issues getting some testing libraries configured correctly. Testing was mostly accomplished via Postman, which worked well and allowed for code changes to be made without also needing to make many changes to the testing setup.

2 Final Timeline and Initial Timeline Comparison

We adhered to our initially laid out sprint schedule by committing to two week sprint cycles, followed by sprint review and sprint planning. Although we fell behind on our overall timeline by not making as much progress in development as we anticipated. Our average velocity for the duration of the project was 17.6 Story Points per Sprint.

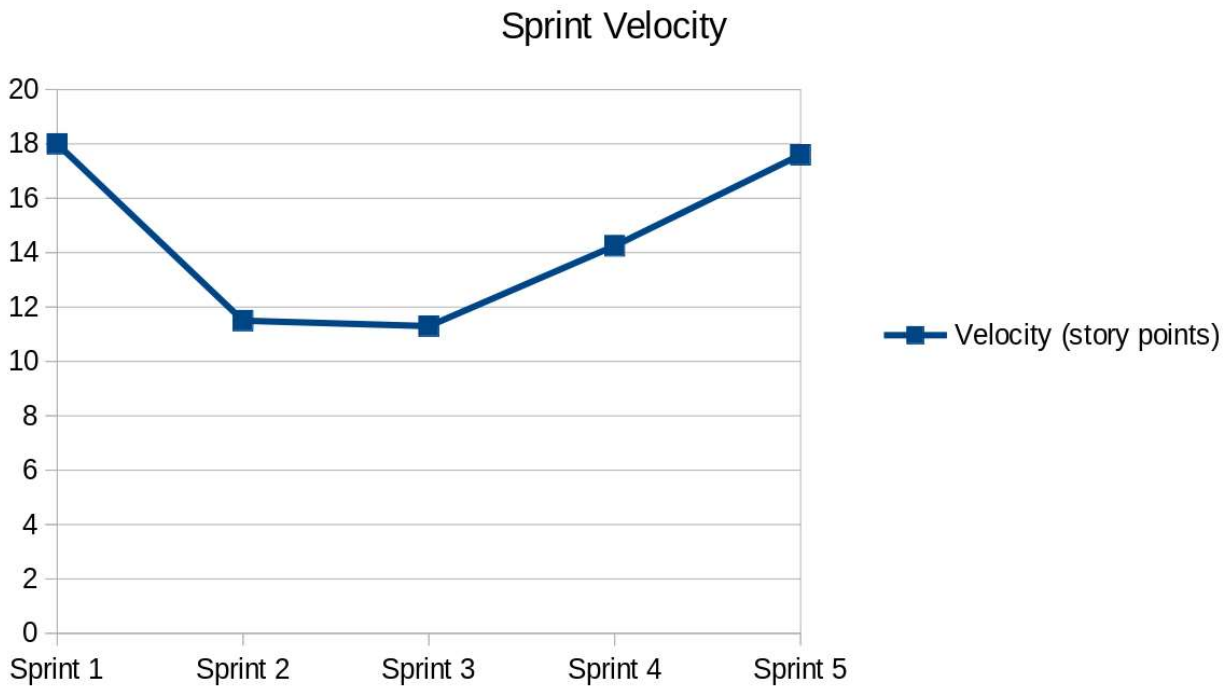

figure 1

3 Project Results Compared With Expectations

Issues with scheduling around our senior semesters lead to delays in the project generally, which resulted in a scaled back Minimum Viable Product. We didn't have a chance to implement our stretch goals of user messaging, and group conferencing. Additionally, our nested comments need work in order to have their implementation fully align with requirements. Generally though, the project is still on track. Assuming development were to continue the amount of tech debt is minimal, and only a minor refactor would be required to get it in shape for another major development push.

4 Software Evaluation

Software will be evaluated according to both its architecture and the security of the application.

4.1 Software Architecture

A fairly loose interpretation of a 3 tier architecture resulted in more business logic being done in the presentation layer than would be preferable, but refactoring it out into services should be relatively painless. Initial designs of the API proved to be relatively stable, with only tweaks done as we progressed with the project.

4.2 Security

In this section I cover the software evaluation from a security perspective, going over the testing of various potential vulnerabilities across the attack surface of Study Island.

4.2.1 Denial of Service (DoS)

DoS attacks traditionally come in the form of mass requests from distributed attackers making resources inaccessible to real users, but can be specially crafted to target an application to exploit poor resource management or poor error handling to accomplish the same goal.

In the case of Study Island, the lack of error handling in certain places means that input fuzzing can cause unrecoverable crashes and exit the program.

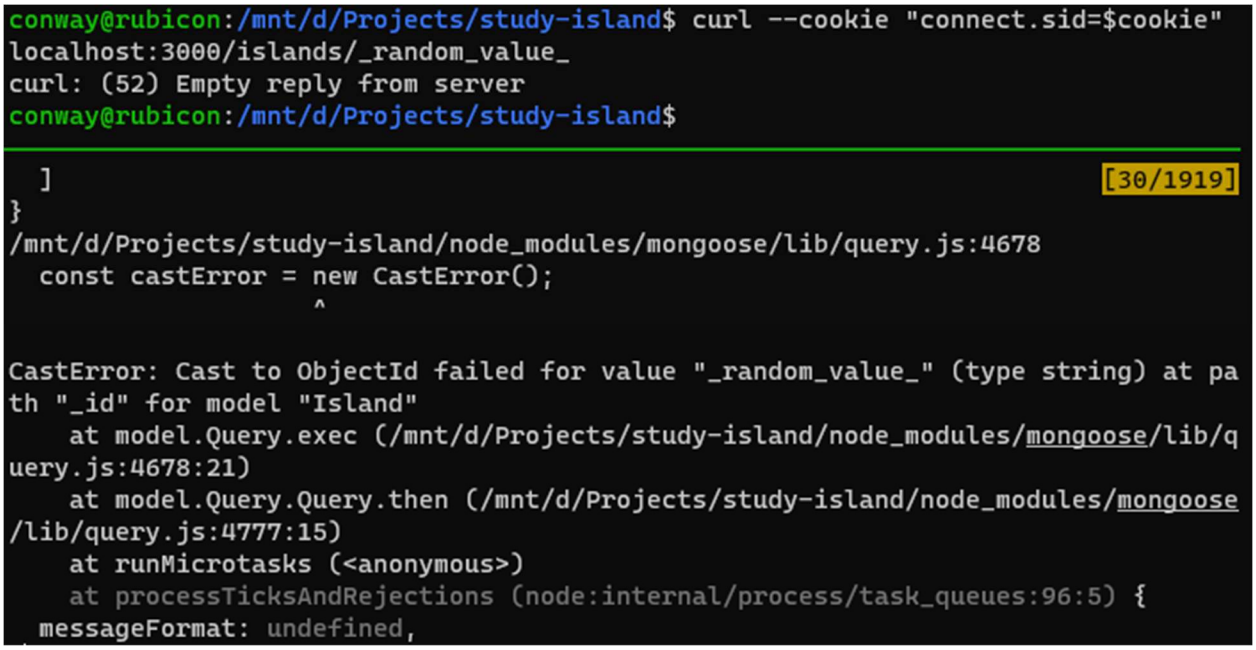

figure 2

4.2.2 Cross-Site Scripting (XSS)

XSS vulnerabilities arise when data is improperly displayed within the client's browser. These can come in the form of reflected XSS (i.e. the server returns your input in a response an an improper way), DOM-based XSS (i.e. the client displays your input in an improper way), stored XSS (i.e. the server returns data from a database and displays it in an improper way). An attacker could use this vulnerability to steal account information or redirect the browser to another malicious page.

Various XSS methods were tested. In the current state of the software, there are no points where DOM-based XSS would be possible. Usernames, island names, threads, replies, and searches were analyzed for stored XSS and reflected XSS vulnerabilities without compromise.

figure 3

4.2.3 NoSQL Injections

Because Study Island utilizes MongoDB as its database, traditional SQL injections do not work, so as such, NoSQL injection techniques need to be utilized to evaluate the security of the database integrations. NoSQL injections were tested in numerous input fields including the login, registration, and searches without compromise.

```
conway@rubicon:/mnt/d/Projects/study-island$ curl -v -X POST -H 'content-type: application/json' -d
'{"username":"{ \"\\$gte\": \"\" }", "password":"{ \"\\$gte\": \"\" }"}' localhost:3000/auth/login
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 127.0.0.1:3000...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 3000 (#0)
> POST /auth/login HTTP/1.1
> Host: localhost:3000
> User-Agent: curl/7.68.0
> Accept: */*
> content-type: application/json
> Content-Length: 70
>
* upload completely sent off: 70 out of 70 bytes
* Mark bundle as not supporting multiuse
< HTTP/1.1 302 Found
< X-Powered-By: Express
< Location: /auth/login
< Vary: Accept
< Content-Type: text/plain; charset=utf-8
< Content-Length: 33
< Date: Fri, 29 Apr 2022 01:44:06 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
* Connection #0 to host localhost left intact
```

figure 4

5 Work To Be Done

At this point we have completed the basic functionality of the application and it is deployable. Currently the back-end is in place for the implementation of Mod and Admin actions, but we haven't had time yet to implement them in the client. Further work needs to be done on Posts as well to allow rendering thumbnails for links, embedding videos, etc. Going forward we will conduct a thorough code review to ensure our code will support the expansion of features we anticipate.

Long term, we would plan to begin implementation of the study group functionality that would allow users to form groups of various sizes and meet in a virtual study environment where they can communicate via voice and text chat, share files with each other, and screen share. We also need to implement the federated islands functionality that will allow two separate islands to join together to share information that they both may have in common and may benefit users from either island.